User-friendly KR with cDMN

A table-based format for KR

Simon Vandevelde, 26/02/2025

Whoami

- Simon Vandevelde
- Post-doctoral researcher at KU Leuven
- DTAI research group
- Focus on **user-friendly KR**

> 💭 **General research**
> How can we make KR more accessible for non-experts? What difficulties do companies experience adopting KR, and how can we overcome them?

→ Research targets *applied KR*

me.

Some examples of past use cases

📝	Notary[1]	Help notaries ask the right questions
⚙️	Component designer[2]	Help engineers design components with less errors
💶	Intelli-Select[3]	Collateral management of large financial assets
🧬	Adhesive Selector[4]	Find suitable adhesives *quickly*

[1] Deryck, M., Devriendt, J., Marynissen, S., & Vennekens, J. (2019). Legislation in the KB paradigm: Interactive decision enactment for registration duties.

[2] Aerts, B., Deryck, M., & Vennekens, J. (2022). Knowledge-based decision support for machine component design: A case study.

[3] Callewaert, B., Decleyre, N., Vandevelde, S., Comenda, N., Coppens, B., & Vennekens, J. (2023). Facilitating Investment Strategy Negotiations through Logic.

[4] Vandevelde, S., Vennekens, J., Jordens, J., Van Doninck, B., & Witters, M. (2024). Knowledge-Based Support for Adhesive Selection: Will it Stick?

Common element

🗯️ What do all of these applications have in common?
Require building extensive KBs through knowledge acquisition!

- Formalizing expert knowledge into formal KB
- Requires both domain expert and knowledge engineer
- Costly, error-prone
- One of the **main challenges for KR**

→ **Knowledge acquisition bottleneck**

🤔 What if domain experts could model by themselves?

But... Teach non-experts KR?

- Learning KR languages requires extensive training
- Many intricacies
- Finicky symbols, syntax
- Domain experts might not have maths/engineering/CS background
- Even for IT: whole new paradigm

💡 Perhaps we could use a different, easier formalism?

DMN

- Decision Model and Notation

- Published by OMG group (like UML)

- Decision logic

- Increasingly high adoption rate in industry, vibrant community

🎯 DMN is designed to be *readable and writeable* for everyone involved in the modelling process, ranging from IT experts to policy makers and …
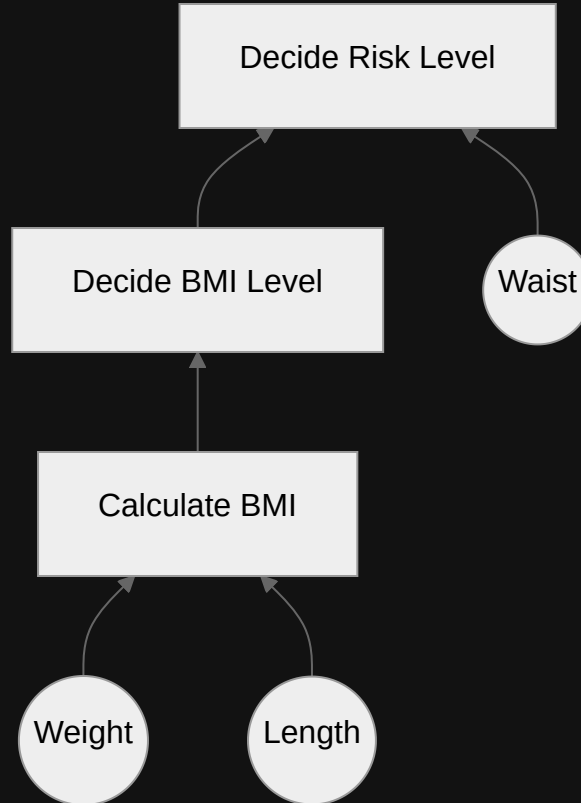
Main building block of DMN:

Decision Tables

Example Decision Table

U	BMI	BMI Level
1	< 18.5	Underweight
2	[18.5..25]	Normal
3	[25..30)	Overweight
4	[30..35)	Obese
5	>= 35	Morbidly Obese

- Input(s), green
- Output, blue
- Each row = "rule"
- Behavior of table: hit policy (U)

$\rightarrow$ simple tables can be "puzzled together" for complex decision making

BMI example

BMI example

Example: 79kg, 1.70m, waist of 113

U	Weight	Length	BMI
1	-	-	Weight/(Length * Length)

U	BMI	BMI Level
1	< 18.5	Underweight
2	[18.5..25]	Normal
3	[25..30)	Overweight
4	[30..35)	Obese
5	>= 35	Morbidly Obese

U	BMI Level	Waist	Risk level
1	Underweight, Normal	-	Low
2	Overweight	< 105	Low
3	Overweight	>= 105	Increased
4	Obese	< 105	Increased
5	Obese	>= 105	High
6	Morbidly Obese	-	High

DMN: benefits

- Readable + writeable by many
- Directly executable: no coding necessary!
  - No coding necessary
  - Avoids coding bugs
  - Avoids misinterpretation errors
  - "Rules as code"!
- Many commercial solvers available

Bijlage I: Tabel Beslissingsregels voor het generen van het Covid Safe Ticket

DMN: limitations

Main limitation: all tables are definitions

→ output values are *defined* by input

→ output cells must always contain precise value

✏️ ❓   "You must have a low health risk to join the army"

U	Risk Level	Army
1	not(Low)	No
2	Low	Yes

U	Army	Risk Level
1	Yes	Low
2	No	???

⊘
DMN offers no good way to express constraints!

cDMN

- *constraint* Decision Model and Notation

- Introduces constraint tables using E^* hit policy

- All rows must hold

- Does not require tables to be complete

E*	Army	Risk Level
1	Yes	Low

Read as "If you wish to join the Army, then your Risk Level must be Low".

$\rightarrow$ This small addition makes many more applications possible!

cDMN: representations

DMN supports constants (e.g., *Risk Level*) and booleans (e.g., *Army*)

$\rightarrow$ cDMN extends these with types, predicates and functions

$\rightarrow$ In the glossary

Consider a doctor planning problem:

Type		
Name	**Type**	**Possible Values**
Doctor	String	Fleming, Freud, Heimlich, Eustachi, Golgi
Day	String	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday

Function	
Name	**Type**
doctor assigned to Day	Doctor

$(assigned/1 : Day \rightarrow Doctor)$

Predicate
Name
Doctor is available on Day

$(available/2 : Doctor \times Day \rightarrow \mathbb{B})$

cDMN: quantification

cDMN allows **universal** quantification by using type as input

✏️ ❓ "A doctor can only be assigned to a Day if they are available"

E*	Doctor	Day	doctor assigned to Day	Doctor is available on Day
1	-	-	Doctor	Yes

$$\forall\, do\, \in\, Doctor : \forall\, da\, \in\, Day : assigned(da) = do \Rightarrow available(do, da).$$
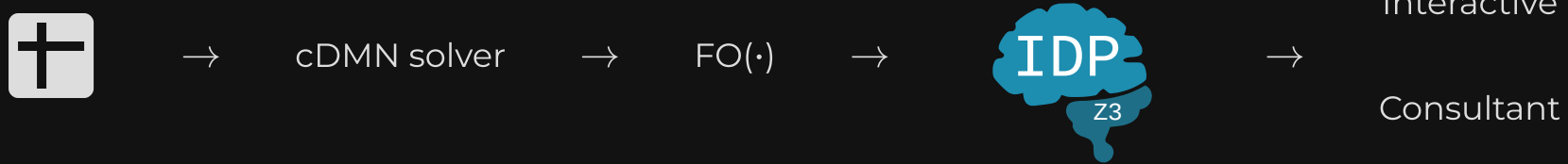
cDMN: datatables

If we already know the interpretation of a symbol, we can use a data table

E.g.:

D	Doctor	Day	Doctor is available on Day
1	Heimlich	Monday, Tuesday, Wednesday	Yes
2	Eustachi	Tuesday, Wednesday, Friday	Yes
3	Freud	Monday, Wednesday, Thursday	Yes
…	…	…	…

How to use the model?

■ → cDMN solver → FO(·) → IDP z3 → Interactive Consultant

Interactive Consultant

- Interface to interact with KB
- Generic, UI is fully automatic
- Highly interactive: effect of each choice is automatically shown

Demo

Extended Doctor Planning

Schedule of 7 days, consisting of three shifts, given 5 doctors and the following rules:

1. A doctor can only work one shift per day

2. A doctor should always be available for their shift

3. If a doctor has the night shift, they either get the next day off or the night shift again

4. A doctor either works both days of the weekend, or none of the days.

5. Golgi works at most two nights.

Doctor Planning in IC

More cDMN examples

cdmn.be contains many examples, mostly from DM Community challenges:

- Planning/scheduling problems: Doctor Planning, Course selection (still to be published)
- Configuration problems: Hamburger Challenge, Miss Manners, Balanced Assignment
- Decision management: Vacation Days Advanced, Covid Testing
- Classic constraint problems: Map Coloring, Map Coloring with Violations

Summing up cDMN

- Table-based KR notation
- Aims to be user-friendly
- Based on popular DMN standard
- Adds constraints, quantification, data tables, and more.
- Models can be solved using IDP-Z3
- Interactive, explainable environment in Interactive Consultant

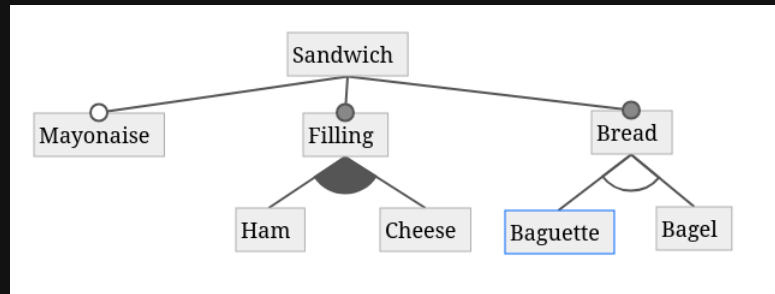In use cases: cDMN seems to work quite well! Formal validation pending.

> 💭 **Just an idea...**
> ASP-based cDMN solver? :-)

Aside

Also other work for user-friendly KR

- Feature models for easy configuration

- Controlled Natural Languages

- Blocks-based interfaces

- Interaction with KB (e.g., LLMs)

- …

Thank you

More info:

- https://cdmn.be
- https://idp-z3.be
- https://interactive-consultant.idp-z3.be
- https://slides.simonvandevelde.be/Potsdam25/slides.pdf

Always open for collaboration! :-)

🌐 https://simonvandevel.de

✉ s.vandevelde@kuleuven.be

🐘 @saltfactory@mastodon.social

in Simon Vandevelde